PES UNIVERSITY
(Established under Karnataka Act No. 16 of 2013)
100 Ft. Road, BSK III Stage, Bengaluru – 560 085
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SESSION: AUG-DEC 2020

Course Title: Algorithms for Information Retrieval		
Course code: UE17CS412		
Semester : VII sem	**Section: A/E/G**	**Team Id:10**
SRN:PES1201700226	**Name: Sparsha P**	
SRN:PES1201700286	**Name: R Ananth**	
SRN:PES1201700298	**Name: Shashank MG**	
SRN:PES1201701557	**Name: Pramod MN**	

ASSIGNMENT REPORT

Problem Statement

- Build a search engine for Environmental News NLP archive.
- Built a corpus for archive with at least 417 documents
- Provide support for querying documents including:
  - Simple query
  - Phrase query
  - Wildcard query
- Provision for comparing the built IR system with state of the art search systems such as Elastic search (ES) or Apache Solr.

Description

We have taken a BTree approach to store the index files for the IR system. We store these precomputed indexes as a .pickle file as they are represented as a HashMap. We have a total of 417 index files being generated. These indexes are positional in nature so as to provide support to phrase queries. Further we also generate a bigram index for the corpus to perform wildcard queries. These indexes also store additional information to help compute the TF-IDF score which is necessary for ranking the documents. The simple query is a mere combination of the retrieve and intersection algorithm. The positional query has a postfilter step to make sure the terms occur in the sequence. The wildcard query too has a postfilter to remove the false positives.

Elastic search too was set up in a similar manner by configuring the mappings for the index. 417 indexes were created to simulate the storage pattern as our IR system, the scoring function used was TF-IDF.

We have written a script to compare ES and our IR system when we retrieve documents from all the indexes. The brief summary is mentioned below.

Code Snippets and Output

1. Index Construction

a. Code Snippet

```
for pos in range(len(listOfWords)):
    lemWord = listOfWords[pos]

    if(lemWord.isalnum()):
        if(invertedIndex.has_key(lemWord)):
            if(docId not in invertedIndex[lemWord][1]):
                invertedIndex[lemWord][1][docId] = [1,[pos]]
                invertedIndex[lemWord][0] += 1
            else:
                invertedIndex[lemWord][1][docId][1].append(pos)
                invertedIndex[lemWord][1][docId][0] += 1
        else:
            postingListElement = [1, {docId : [ 1, [pos]]}]
            invertedIndex.insert(lemWord, postingListElement)
```

2. Simple Query

a. Code Snippet

```
def searchOnlyTerms(searchTerms, invertedIndex, topK, N):
    rankDict = {}
    searchTerms = getTokens(searchTerms)      sparsha-p, 14 days ago • Positional Search
    for term in searchTerms:
        if(invertedIndex.has_key(term)):
            IDF = math.log(N/invertedIndex.get(term)[0] ,10)
            for docIDforTerm in invertedIndex[term][1]:
                if(docIDforTerm in rankDict):
                    rankDict[docIDforTerm] += invertedIndex.get(term)[1][docIDforTerm][0] * IDF
                else:
                    rankDict[docIDforTerm] = invertedIndex.get(term)[1][docIDforTerm][0] * IDF
        else:
            pass
```

b. Output :

- Search Phrase

```
{
    "query":
    {
        "mode":1,
        "search" : ["greenhouse","gas"],
        "top":5
    }
}
```

- Search on all files

```
HP@RANANTH MINGW64 /d/PES-UNIVERSITY/AIR/InformationRetrieval/src (main)
$ python queryDriver.py

Time taken by IR : 10.934466361999512
{
 "1": {
  "docName": "MSNBC.201908_1",
  "score": 7.530233181693728,
  "document": {
   "URL": "https://archive.org/details/MSNBCW_20190830_010000_The_Rachel_Maddow_Show#start/641/end/676",
   "MatchDateTime": "8/30/2019 1:10:56",
   "Station": "MSNBC",
   "Show": "The Rachel Maddow Show",
   "IAShowID": "MSNBCW_20190830_010000_The_Rachel_Maddow_Show",
   "IAPreviewThumb": "https://archive.org/download/MSNBCW_20190830_010000_The_Rachel_Maddow_Show/MSNBCW_20190830_010000_The_Rachel_Maddow_Show.thumbs/MSNBCW_2
0190830_010000_The_Rachel_Maddow_Show_000630.jpg",
   "Snippet": "alone were equivalent to greenhouse gas emissions from more than 69 million cars. methane emissions just from oil and gas companies are the equ
ivalent to greenhouse gas to"
  }
 },
 "10": {
  "docName": "MSNBC.201908_10",
  "score": 7.530233181693728,
  "document": {
   "URL": "https://archive.org/details/MSNBCW_20190829_130000_MSNBC_Live_With_Stephanie_Ruhle#start/526/end/561",
   "MatchDateTime": "8/29/2019 13:09:01",
   "Station": "MSNBC",
   "Show": "MSNBC Live With Stephanie Ruhle",
   "IAShowID": "MSNBCW_20190829_130000_MSNBC_Live_With_Stephanie_Ruhle",
   "IAPreviewThumb": "https://archive.org/download/MSNBCW_20190829_130000_MSNBC_Live_With_Stephanie_Ruhle/MSNBCW_20190829_130000_MSNBC_Live_With_Stephanie_Ruh
le.thumbs/MSNBCW_20190829_130000_MSNBC_Live_With_Stephanie_Ruhle_000509.jpg",
   "Snippet": "is the biggest reason why we have been reducing greenhouse gases in this country over the last seven years or so? it is because we have replace
d coal factories with natural gas factories. it's better for the greenhouse gas environment to do that. so there is a trade-off involved"
  }
 },
```

```
----------CONFUSION MATRIX------------
[4, 0]
[1, 0]
-----------------METRICS-----------------
Accuracy : 0.8
Precision : 1.0
Recall : 0.8
FScore : 0.888888888888889
```

3. Phrase Query

a. Code Snippet

```
def postFilter(docsToConsider, allDocList):
    docsToConsider = list(docsToConsider)

    positions = []
    for doc in range(len(docsToConsider)):
        positions.append({})
        for ithWord in range(len(allDocList)):
            if(docsToConsider[doc] in positions[doc]):
                positions[doc][docsToConsider[doc]].append(allDocList[ithWord][1][docsToConsider[doc]][1])
            else:
                positions[doc][docsToConsider[doc]] = [allDocList[ithWord][1][docsToConsider[doc]][1]]
```

b. Output

- Search Phrase

```
{
    "query":
    {
        "mode":1,
        "must" : "I do believe in climate change",
        "top":5
    }
}
```

- Search Result on all Files

```
HP@RANANTH MINGW64 /d/PES-UNIVERSITY/AIR/InformationRetrieval/src (main)
$ python queryDriver.py
operation took: 11.018142700195312
{
 "45": {
  "docName": "BBCNEWS.201710_45",
  "score": 3,
  "document": {
   "URL": "https://archive.org/details/BBCNEWS_20171022_223000_The_Papers#start/746/end/781",
   "MatchDateTime": "10/22/2017 22:42:41",
   "Station": "BBCNEWS",
   "Show": "The Papers",
   "IAShowID": "BBCNEWS_20171022_223000_The_Papers",
   "IAPreviewThumb": "https://archive.org/download/BBCNEWS_20171022_223000_The_Papers/BBCNEWS_20171022_223000_The_Papers.thumbs/BBCNEWS_20171022_223000_The_Papers_000718.jpg",
   "Snippet": "on the way, you are going down the climate change. i do believe in climate change. i do believe in climate change. i do believe in climate change but what i think is interesting is when it starts happening in the way it's been happening in the way it's been happening in the west, notjust here but america, i think a lot of people"
  }
 }
}
{
 "41": {
  "docName": "CNN.201706_41",
  "score": 1,
  "document": {
   "URL": "https://archive.org/details/CNNW_20170602_190000_CNN_Newsroom_With_Brooke_Baldwin#start/249/end/284",
   "MatchDateTime": "6/2/2017 19:04:24",
   "Station": "CNN",
   "Show": "CNN Newsroom With Brooke Baldwin",
   "IAShowID": "CNNW_20170602_190000_CNN_Newsroom_With_Brooke_Baldwin",
   "IAPreviewThumb": "https://archive.org/download/CNNW_20170602_190000_CNN_Newsroom_With_Brooke_Baldwin/CNNW_20170602_190000_CNN_Newsroom_With_Brooke_Baldwin.thumbs/CNNW_20170602_190000_CNN_Newsroom_With_Brooke_Baldwin_000238.jpg",
   "Snippet": "it will be fascinating if he finally says i do believe in climate change because that would mean in the past he had been wrong on the issue. there's been so much dancing around in recent days, but even during the campaign he seemed to soften his stance when asked about whether climate change was"
  }
 }
}
```

4. Wildcard Query

a. Code Snippet

```
def generateBiGramsForQuery(query):
    query = splitQuery("$"+query+"$")
    toReturn = []
    for i in range(len(query)):
        bigramKey = list(bigrams(list(pad_sequence(query[i],n=2))))
        toReturn.extend(bigramKey)
    return toReturn

def getBiGrams(bigramIndex, listOfBigrams):
    toRet = []
    for bigram in listOfBigrams:
        toRet.append(bigramIndex.get(bigram))
    return toRet

def intersectionBiGrams(possibleWords):
    possibleWords = sorted(possibleWords, key= lambda x:len(x))
    return(list(reduce(lambda x,y: set(x) & set(y), possibleWords)))

def postFilter(regex, listOfCandi):
    regex = regex.replace("*", ".*")
    return list(filter(lambda x : re.search(regex, x), listOfCandi))

def wordRetrieval(query, bigramIndex):
    biGramsForQuery = generateBiGramsForQuery(query)
    if(biGramsForQuery):
        biGramLists = getBiGrams(bigramIndex,biGramsForQuery)
        words = intersectionBiGrams(biGramLists)
        filterWords = postFilter(query,words)
        return(filterWords)
    return([])
```

b. Output

- Search Phrase

```
{
    "query":
    {
        "mode":1,
        "wildcard" : "re*",
        "top":5
    }
}
```

- Search Results

```
HP@RANANTH MINGW64 /d/PES-UNIVERSITY/AIR/InformationRetrieval/src (main)
$ python queryDriver.py
operation took: 6.543658018112183
{
 "489": {
  "docName": "BBCNEWS.201905_489",
  "score": 6,
  "document": {
   "URL": "https://archive.org/details/BBCNEWS_20190510_100000_BBC_Newsroom_Live#start/1199/end/1234",
   "MatchDateTime": "5/10/2019 10:20:14",
   "Station": "BBCNEWS",
   "Show": "BBC Newsroom Live",
   "IAShowID": "BBCNEWS_20190510_100000_BBC_Newsroom_Live",
   "IAPreviewThumb": "https://archive.org/download/BBCNEWS_20190510_100000_BBC_Newsroom_Live/BBCNEWS_20190510_100000_BBC_Newsroom_Live.thumbs/BBCNEWS_20190510
_100000_BBC_Newsroom_Live_001197.jpg",
   "Snippet": "radical new ways to try to repair the climate and reverse global warming are being considered by scientists at the university of cambridge as p
art of plans for a new research centre. among the ideas is a scheme to re-freeze polar regions, by reflecting sunlight away from the earth and spraying water
from ships into the atmosphere. 0ur science correspondent, pallab ghosh, reports."
  }
 }
}
{
 "485": {
  "docName": "BBCNEWS.201905_485",
  "score": 6,
  "document": {
   "URL": "https://archive.org/details/BBCNEWS_20190510_050000_Breakfast#start/210/end/245",
   "MatchDateTime": "5/10/2019 5:03:45",
   "Station": "BBCNEWS",
   "Show": "Breakfast",
   "IAShowID": "BBCNEWS_20190510_050000_Breakfast",
   "IAPreviewThumb": "https://archive.org/download/BBCNEWS_20190510_050000_Breakfast/BBCNEWS_20190510_050000_Breakfast.thumbs/BBCNEWS_20190510_050000_Breakfas
t_000207.jpg",
   "Snippet": "-- ramzan karmali, bbc news. radical new ways to repair the climate and reverse global warming are being considered by scientists at the univer
sity of cambridge as part of a new research centre. among the ideas it will consider is a scheme to re-freeze polar regions, by reflecting sunlight away from
the earth, using water sprayed into the atmosphere by ships. this report by our science"
  }
```

Interpretation of efficiency

- We see that when we are retrieving for a single file we fetch the documents fetched by ES.
- When we are querying over all the indexes we can observe about 70% accuracy which increases as the K value increases. This is mostly because we are not performing normalization in our IR system.
- The key difference and the most important one is the query response time. ES responds in about 50ms whereas our IR system takes 4-5 seconds for a query on all the indexes.

K	Accuracy	Precision	Recall	F1-score	Time (ES)	Time (IR)
5	0.8	1	0.8	0.8889	1.778	9.713
25	0.72	1	0.72	0.83	2.134	11.292
50	0.62	1	0.62	0.765	4.408	9.587
100	0.62	1	0.62	0.765	2.730	10.011

Learning Outcome

- This assignment exposed us to real world search engines like ElasticSearch.
- We also learnt how to implement the learnt algorithms in practical applications.
- We were introduced to various libraries and methodologies which would otherwise not be explored
- The efficiency comparison also showcased how important optimization is.

Name and Signature of the Faculty